

## **REMARKS**

Claims 1-49 were examined. Applicant has amended claims 1, 6, 38 and 43. Claim 49 is cancelled. No claims are newly presented. No new matter has been introduced.

### **Amendments to the Drawings:**

Applicants have amended the drawings in compliance with the Examiner's recommendations. Applicants note that the Examiner refers to Figure 3 as Figure 2 and Figure 4 as Figure 3 in his objections.

### **Objections to the Specification**

The specification is objected to because of formalities. Applicant has amended the specification.

### **Claim Objections**

Claims 1 and 6 are objected to due to informalities. Applicant has amended the claims to overcome this ground of rejection.

### **Rejections under 35 U.S.C. §101**

Claim 49 is rejected under §101 as directed to non-statutory subject matter. Applicant has amended claim 49. Claim 49 has been canceled and this ground of rejection is now moot.

### **Rejections under 35 U.S.C. §102**

Claims 1, 38 and 43 stand rejected under §102(e) as anticipated by "WinRunner 7.0 Tutorial".

This ground of rejection is respectively traversed.

### **Rejections under 35 U.S.C. §103**

Claims 1-48 stand rejected under §103(a) as being obvious over Melamed et al. 9US 2004/0107415 in view of WinRunner.

This ground of rejection is respectively traversed.

In one embodiment of the present invention, as set forth in claim 1, a method is provided for transforming test cases that are converted to an abstract representation. Test cases are imported that are written in one or more scripting languages. Semantic analysis is used to convert test cases to an abstract representation that includes application state, external interaction sequences and input data. The application state is a set of application objects associated with a set of attributes and their values, or represents a runtime snapshot of an application under test which defines a context of external interaction. Environment mappings are used to provide platform independence of test cases. The abstract representation provides a platform independent representation of test case. The abstract representation of test cases is stored in a database system. Test scripts are generated for multiple test execution environments.

WinRunner creates tool dependency and reduces ability to share the tests. Additionally, WinRunner does not use semantic analysis to convert test cases to abstract representations, and test data is not separated from test steps in abstract representations.

WinRunner supports two modes of development of test scripts: manual script development and recording of user interactions. In WinRunner the tool records user interactions into a script form which can be played back to execute the tests automatically without manual intervention. One of the disadvantages of this approach is that the scripts generated through recording are too fragile and requires additional manual efforts to ensure reusability for parameterization. Another disadvantage is that the scripts created using one tool will not run on another tool. This reduces the ability to share the test scripts with customer or vendors if they use a different tool.

The present invention specifically addresses the disadvantages of the record and playback tool of WinRunner. With the present invention, the test scripts are converted to a platform independent format (abstract representation) and the test cases can be shared seamlessly. Test scripts can be rebuild from abstract representation for multiple platforms, and the test cases can be executed in any platform from any vendor.

WinRunner uses tool specific scripting as compared to the present invention that uses abstract representation. WinRunner has its own specific scripting language for recording and playback.

This is clearly distinguished from the present invention where test scripts are converted to an abstract representation during the import, and then the test scripts can be generated in multiple languages. we remove tool dependency and increases reuse.

With WinRunner, test scripts contain test data embedded in test steps. Parameterization is done manually to remove data dependency and enable scripts to run with multiple test data. With the present invention, test data is separated from test steps in abstract representations. This makes it easy to run same test cases with multiple data sets.


With WinRunner, application states, external interaction sequences (test steps) and data are mixed together within the scripts. It is very difficult to reuse or modify the test cases later as a result. This is distinguished from the present invention which uses abstract representations to distinguish these entities.

### CONCLUSION

It is submitted that the present application is in form for allowance, and such action is respectfully requested. The Commissioner is authorized to charge any additional fees which may be required, including petition fees and extension of time fees, to Deposit Account No. 08-1641 (Docket No. 07464-0002).

Respectfully submitted,  
HELLER EHRMAN LLP

Date: 11.29.2007

  
Paul Davis, Reg. No. 29,294

275 Middlefield Road  
Menlo Park, CA 94205  
(650) 324-7041  
Customer No. 25213